

# The Complexity of Approximating a Bethe Equilibrium

Jinwoo Shin\*

November 17, 2011

## Abstract

This paper resolves a common complexity issue in the Bethe approximation of statistical physics and the sum-product Belief Propagation (BP) algorithm of artificial intelligence. The Bethe approximation reduces the problem of computing the partition function in a graphical model to that of solving a set of non-linear equations, so-called the Bethe equation. On the other hand, the BP algorithm is a popular heuristic method for estimating marginal distributions in a graphical model. Although they are inspired and developed from different directions, Yedidia, Freeman and Weiss (2004) established a close connection: the BP algorithm solves the Bethe equation if it converges (however, it often does not). This naturally motivates the following important question to understand their limitations and empirical successes: the Bethe equation is computationally easy to solve?

We present a message passing algorithm solving the Bethe equation in polynomial number of operations for arbitrary binary graphical models of  $n$  variables where the maximum degree in the underlying graph is  $O(\log n)$ . Our algorithm, an alternative to BP fixing its convergence issue, is the first fully polynomial-time approximation scheme for the BP fixed point computation in such a large class of graphical models, while the approximate fixed point computation is known to be (PPAD-)hard in general. We believe that our technique is of broader interest to understand the computational complexity of the cavity method in statistical physics.

## 1 Introduction

In the recent years, graphical models (also known as Markov random fields) defined on graphs have been studied as powerful formalisms modeling inference problems in numerous areas including computer vision, speech recognition, error-correcting codes, protein structure, networking, statistical physics, game theory and combinatorial optimization. Two central problems, commonly addressed in these applications involving graphical models, are computing the marginal distribution and the so-called partition function. It is well-known that the inference problems are computationally hard in general [5]. Due to such a theoretical barrier, efforts have been made to develop heuristic methods.

The sum-product Belief Propagation (BP) algorithm (first proposed by Pearl [19]) and its variants (e.g. Survey Propagation) are such heuristics, driven by certain experimental thoughts, for computing the marginal distribution. Their appeals lie in the ease of implementation as well

---

\*Algorithms and Randomness Center at Georgia Institute of Technology, Atlanta GA 30332. Email: jshin72@cc.gatech.edu.

as optimality in tree-structured graphical models (models which contain no cycles). BP (and message-passing algorithms in general) can be thought as an updating rule on a set of messages:

$$m^{t+1} = f(m^t),$$

where  $m^t$  is the multi-dimensional vector of messages at the  $t$ -th iteration, and  $f$  describes the updating rule (or BP operator). Two major hurdles to understand such a message-passing algorithm are about its convergence (i.e.  $m^t$  converges to  $m^*$ ?) and correctness (i.e.  $m^*$  is good enough?). It is known that the BP iterative procedure always has a fixed point  $m^*$  due to the Brouwer fixed point theorem. However, BP can oscillate far from a fixed point in models with cycles, and only several sufficient convergence conditions [25, 22, 14, 15] have been established in the last decade. More importantly, BP can have multiple fixed points, and even when it is unique, it may not be the correct answer. Significant efforts [14, 24, 26] were made to understand BP fixed points, while the precise approximation qualities and the rigorous understandings on their limitations still remain mystery. Regardless of those theoretical understandings, the BP algorithm performs empirically well in many applications [12, 18, 10]. For example, the highly successful turbo codes [3] in practice can be interpreted as BP [17] and decisions guided by BP is also known to work well to solve satisfiability problems [20].

The Bethe approximation [2, 26] and its variants (e.g. Kikuchi approximation [9]), originally developed in statistical physics of lattice models, are currently used as powerful approximation schemes for computing the (logarithm of) partition function in many applications. The Bethe approximation suggests to use the following quantity as an approximation for the logarithm of the partition function:

$$F(\mathbf{y}^*) \quad \text{where} \quad \nabla F(\mathbf{y}^*) = 0.$$

$F$ ,  $\nabla F(\mathbf{y}^*) = 0$  and  $\mathbf{y}^*$  are called the (minus) Bethe free energy function, Bethe equation and Bethe equilibrium, respectively. The statistical physics prediction suggests its asymptotic correctness in random sparse graphical models, and several rigorous evidences in particular models are known [1, 8, 4]. Efforts have also been made to estimate and characterize its error [6, 21]. However, the error still remains uncontrollable for models with many cycles.

Yedidia, Freeman and Weiss [26] established a somewhat surprising connection between the BP algorithm and the Bethe approximation: if BP converges, it solves the Bethe equation. Equivalently, the BP fixed point equation  $f(m^*) = m^*$  is in essence equivalent to the Bethe equation  $\nabla F(\mathbf{y}^*) = 0$ . This naturally leads to the following common computational question for both: the BP fixed point computation is computationally easy? Formally speaking,

*Q.* Given  $\varepsilon > 0$ , is it possible to design a deterministic iterative algorithm finding  $m^*$  satisfying<sup>1</sup>

$$(1 - \varepsilon) f(m^*) \leq m^* \leq (1 + \varepsilon) f(m^*),$$

in polynomial number of bitwise operations with respect to  $1/\varepsilon$  and the dimension of vector  $m^*$ ?

Such an algorithm can be used an alternative to BP with provably fast convergence rate (i.e. fixing the convergence issue of BP) and eliminates a need for the convergence analysis of BP. Even though it may not converge to the correct answer, it can, at least, provide a guidance

---

<sup>1</sup>Note that it is impossible to compute the exact solution  $m^*$  with  $f(m^*) = m^*$  since it is irrational in general.

toward it [20]. Further, it confirms the efficiency of the Bethe approximation scheme as well since  $m^*$  satisfying the above inequality provides  $\mathbf{y}^*$  with  $\|\nabla F(\mathbf{y}^*)\| \leq \varepsilon$ . Efforts to design such algorithms were made [23, 27], but no rigorous analysis on their convergence rates is known. The authors in [4] provide an algorithm with provable polynomial convergence rate, but the work is for a very specific graphical model (i.e. the uniform distributions on independent sets of sparse graphs). It is far from being clear whether such a poly-convergence algorithm exists for general graphical models. This is primarily because computing a fixed point and a local minimum approximately are not known to be believably (PPAD- or PLS-)hard in general [7].

## 1.1 Our Contribution

The main result of this paper is the following answer  $\mathcal{A}$  (cf. Theorem 2 in Section 3) for the question  $\mathcal{Q}$  for the BP operator  $f$  and arbitrary sparse binary graphical models. To state it formally, we let  $n$  be the number of nodes and  $\Delta$  be the maximum degree in the underlying graph, respectively.

$\mathcal{A}$ . Given  $\varepsilon > 0$ , there exists a deterministic iterative algorithm finding  $m^*$  satisfying

$$(1 - \varepsilon) f(m^*) \leq m^* \leq (1 + \varepsilon) f(m^*)$$

in  $2^{O(\Delta)} n^2 \varepsilon^{-4} \log^3(n \varepsilon^{-1})$  iterations.

In this paper, we call the message  $m^*$  satisfying the above inequality as an  $\varepsilon$ -approximate BP fixed point. In what follows, we explain the algorithm in details.

The known equivalence [26] between the BP fixed point equation and the Bethe equation implies that the question  $\mathcal{Q}$  is equivalent to the following.

$\mathcal{Q}'$ . Given  $\varepsilon > 0$ , is it possible to design a deterministic iterative algorithm finding  $\mathbf{y}^*$  satisfying

$$\|\nabla F(\mathbf{y}^*)\| \leq \varepsilon,$$

in polynomial number of bitwise operations with respect to  $1/\varepsilon$  and the dimension of the domain  $D$  of the Bethe free energy function  $F$ ?

However, we remind the reader that it is still far from being obvious whether it is computationally ‘easy’ to find such a near stationary point or an approximate local minimum (or maximum). Natural attempts are gradient-descent algorithms to find a local minimum or maximum of  $F$ : iteratively update  $\mathbf{y}(t)$  as

$$\mathbf{y}(t+1) = \mathbf{y}(t) + \alpha \nabla F(\mathbf{y}(t)),$$

where  $\alpha \in \mathbb{R}$  is the (appropriately chosen) step-size. The main issue here is that the gradient-descent algorithm may not find a near stationary point if  $\mathbf{y}(t)$  hits the boundary of  $D$  in one of its iterations (and projection is required). Hence, the main strategy in [4] to avoid the hitting issue lies in (a) understanding the behavior of gradient  $\nabla F$  close to the boundary of  $D$  and (b) designing an appropriate small step-size in the gradient-descent algorithm based on the understanding (a).

The main technical challenge to apply the strategy to general binary graphical model (beyond the specific model in [4]) is on (a). The domain  $D$  is simply  $[0, \frac{1}{2}]^n$  in [4] since the Bethe free energy function  $F$  is determined by node marginal probabilities in the uniform independent-set model. One can observe that the proof strategy in [4] is very sensitive and immediately fails even

for the non-uniform independent-set model, which has the domain  $D = [0, 1]^n$ . Furthermore, the more significant issue is that in general graphical model the domain complexity becomes larger, i.e.  $D = [0, 1]^{n+m}$  where  $m$  is the number of edges in the underlying graph. This is because the Bethe free energy should consider pairwise (or edge) marginal probabilities as well. One can check that any similar approaches with [4] fail in the larger domain  $D = [0, 1]^{n+m}$ . To overcome such a technical issue, we first observe that at stationary points of  $F$ , pairwise marginal probabilities should satisfy certain quadratic equations in terms of node marginal probabilities in binary graphical models. This allows to express the Bethe free energy again in terms of node marginal probabilities i.e.  $D = [0, 1]^n$ . Now we study this ‘modified’ Bethe expression  $F^*$  to avoid the hitting issue, which we end up with an appropriate small step-size in the gradient-descent algorithm. Moreover, we eliminate a need to decide such a small step-size explicitly in the algorithm, by designing an elegant time-varying projection scheme.

We later realize that the ‘modified’ Bethe expression  $F^*$  was already proposed by Teh and Welling [23], where they suggested gradient algorithms to minimize  $F^*$  using sigmoid functions. The main difference in our work is that we study the behavior of gradient  $\nabla F^*$  close to the boundary of its domain and guarantee that the gradient-descent algorithm does not hit the boundary without sigmoid functions. The success of our rigorous convergence rate analysis, which was missing in the work of Teh and Welling (2001), primarily relies on this difference. It is also crucial to extend the algorithm design to non-binary graphical models as we describe in Section 4.

One can observe that our gradient-descent algorithm is implementable as a ‘BP-like’ iterative, message passing algorithm: each node maintains a message at each iteration and passes it to its neighbors. We prove it terminates in  $2^{O(\Delta)} n^2 \varepsilon^{-4} \log^3(n \varepsilon^{-1})$  iterations for binary graphical models until it finds an  $\varepsilon$ -approximate BP fixed point. In a complexity point of view, the only remaining issue is that each node may require to maintain irrational messages (of infinitely long bits). We further show that a polynomial number (with respect to  $1/\varepsilon$ ,  $n$  and  $2^\Delta$ ) of bits to approximate each message suffices, and hence the algorithm consists of only a polynomial number of bitwise operations in total. Namely, it is a fully polynomial-time approximation scheme (FPTAS) to compute an approximate BP fixed point for sparse binary graphical models where  $\Delta = O(\log n)$ .

## 1.2 Organization

In Section 2, we provide backgrounds for graphical models, Belief Propagation and Bethe approximation. In Section 3, we describe our algorithm and its time complexity for binary graphical models. In Section 4, we discuss how to extend the result to non-binary graphical models at a high level. From our discussion in Section 4, one can observe that it is not hard to obtain the similar convergence rate result as well. But, we omit the further details in this paper.

## 2 Graphical Models

We first introduce a class of joint distributions defined with respect to (undirected) graphs, which are called (pairwise) *Markov random fields* (MRFs) [16]. Specifically, let  $G = (V, E)$  be a undirected graph with the vertices being denoted by  $V$  with  $|V| = n$ , and the edges  $E \subseteq \binom{V}{2}$  denoting a set of unordered pairs of vertices. The vertices of  $G$  label a collection of random variables  $\mathbf{x} = \{x_v \mid v \in V\}$ . Our primary focus in this paper is on binary random variables, i.e.,  $x_v \in \{0, 1\}$  for all  $v \in V$ .

Now consider the following joint distribution on  $\{0, 1\}^n$  that factors according to  $G$ :

$$p(\mathbf{x}) = \frac{1}{Z} \prod_{v \in V} \psi_v(x_v) \prod_{(u,v) \in E} \psi_{u,v}(x_u, x_v) \quad \text{for } \mathbf{x} \in \{0, 1\}^n.$$

Here, each  $\psi_{u,v}$  and  $\psi_v$  are non-negative functions on  $\{0, 1\}^2$  and  $\{0, 1\}$ , respectively. These local functions are called *potential* functions or *compatibility* functions. The normalizing factor  $Z$  is called the *partition* function:

$$Z = \sum_{\mathbf{x} \in \{0,1\}^n} \prod_{v \in V} \psi_v(x_v) \prod_{(u,v) \in E} \psi_{u,v}(x_u, x_v). \quad (1)$$

Finally, some notations. Let  $\mathcal{N}(v)$  be the set of neighbors of a vertex  $v \in V$ ,  $d_v := |\mathcal{N}(v)|$  be the degree of  $v \in V$ , and  $\Delta := \max_v d_v$  be the maximum degree in the graph  $G$ . Further, we define

$$\psi_* := \max_{(u,v) \in E, x_u, x_v \in \{0,1\}} \left\{ e^{|\ln \psi_v(x_v)|}, e^{|\ln \psi_{u,v}(x_u, x_v)|} \right\}.$$

In this paper, we primarily focus on the case  $\psi_* = O(1)$ .<sup>2</sup>

## 2.1 Belief Propagation

The BP algorithm has messages  $\{m_{u \rightarrow v}^t(\cdot), m_{v \rightarrow u}^t(\cdot)\} := \{m_{u \rightarrow v}^t(x_v), m_{v \rightarrow u}^t(x_u) : (u, v) \in E, x_v, x_u \in \{0, 1\}\}$  at the  $t$ -th iteration are on the both sides of edges and it updates them as

$$m_{u \rightarrow v}^{t+1}(x_v) \propto \sum_{x_u \in \{0,1\}} \psi_{u,v}(x_u, x_v) \psi_u(x_u) \prod_{w \in \mathcal{N}(u) \setminus v} m_{w \rightarrow u}^t(x_u),$$

where  $\sum_{x_v \in \{0,1\}} m_{u \rightarrow v}^{t+1}(x_v) = 1$ . This is equivalent to the following updating rule on (reduced) messages  $\{m_{u \rightarrow v}^t, m_{v \rightarrow u}^t\}$ .

$$m_{u \rightarrow v}^{t+1} = f_{u \rightarrow v} \left( \prod_{w \in \mathcal{N}(u) \setminus v} m_{w \rightarrow u}^t \right),$$

where  $m_{u \rightarrow v}^t := m_{u \rightarrow v}^t(1)/m_{u \rightarrow v}^t(0)$  and the function  $f_{u \rightarrow v} : \mathbb{R}_+ \rightarrow \mathbb{R}_+$  is defined as

$$f_{u \rightarrow v}(x) := \frac{\psi_{u,v}(0, 1)\psi_u(0) + \psi_{u,v}(1, 1)\psi_u(1) \cdot x}{\psi_{u,v}(0, 0)\psi_u(0) + \psi_{u,v}(1, 0)\psi_u(1) \cdot x}.$$

Now the *BP fixed point* of messages  $\{m_{u \rightarrow v}, m_{v \rightarrow u}\}$  can be naturally defined as

$$m_{u \rightarrow v} = f_{u \rightarrow v} \left( \prod_{w \in \mathcal{N}(u) \setminus v} m_{w \rightarrow u} \right),$$

where one can easily argue the existence of such a fixed point using the Brouwer fixed point theorem. This motivates the following notion of  $\varepsilon$ -approximate BP fixed point.

---

<sup>2</sup>This excludes the case  $\psi_{u,v}(\cdot, \cdot) = 0$ . However, we note that our algorithm and its analysis still work even for the case  $\psi_{u,v}(\cdot, \cdot) = 0$  such as the independent set model in [4].

**Definition 1** The set of messages  $\{m_{u \rightarrow v}, m_{v \rightarrow u} : (u, v) \in E\}$  is called an  $\varepsilon$ -approximate BP fixed point if

$$\left| \frac{m_{u \rightarrow v}}{f_{u \rightarrow v} \left( \prod_{w \in \mathcal{N}(u) \setminus v} m_{w \rightarrow u} \right)} - 1 \right| \leq \varepsilon, \quad \forall (u, v) \in E. \quad (2)$$

The BP estimates for node and edge marginal probabilities based on messages, denoted by  $\tau_v(\cdot), \tau_{u,v}(\cdot)$  for  $v \in V, (u, v) \in E$ , are defined as

$$\tau_v(x_v) \propto \psi_v(x_v) \prod_{u \in \mathcal{N}(v)} m_{u \rightarrow v}(x_v) \quad (3)$$

$$\tau_{u,v}(x_u, x_v) \propto \psi_u(x_u) \psi_v(x_v) \psi_{u,v}(x_u, x_v) \prod_{w \in \mathcal{N}(u) \setminus v} m_{w \rightarrow u}(x_u) \prod_{w \in \mathcal{N}(v) \setminus u} m_{w \rightarrow v}(x_v), \quad (4)$$

where  $\sum_{x_v} \tau_v(x_v) = 1$  and  $\tau_v(x_v) = \sum_{x_u} \tau_{u,v}(x_u, x_v)$ .

## 2.2 Bethe Approximation

The Bethe approximation [26] is an approximation to the logarithm of the partition function (i.e.  $\ln Z$ ), given by

$$\sum_{v \in V} \sum_{x_v} \tau_v(x_v) [\ln \psi_v(x_v) - \ln \tau_v(x_v)] + \sum_{\{u,v\} \in E} \sum_{x_u, x_v} \tau_{u,v}(x_u, x_v) \left[ \ln \psi_{u,v}(x_u, x_v) - \ln \frac{\tau_{u,v}(x_u, x_v)}{\tau_u(x_u) \tau_v(x_v)} \right].$$

Under the constraints  $\sum_{x_v} \tau_v(x_v) = 1$  and  $\tau_v(x_v) = \sum_{x_u} \tau_{u,v}(x_u, x_v)$ , it can be reduced as a function of  $\mathbf{y} = [y_v, y_{u,v}]$  where  $y_v = \tau_v(1)$  and  $y_{u,v} = \tau_{u,v}(1, 1)$ .

$$\begin{aligned} F(\mathbf{y}) &:= \sum_{v \in V} y_v (\ln \psi_v(1) - \ln y_v) + (1 - y_v) (\ln \psi_v(0) - \ln(1 - y_v)) \\ &\quad + \sum_{(u,v) \in E} \left[ (1 - y_u - y_v + y_{u,v}) \left( \ln \psi_{u,v}(0, 0) - \ln \frac{1 - y_u - y_v + y_{u,v}}{(1 - y_u)(1 - y_v)} \right) \right. \\ &\quad \quad + (y_u - y_{u,v}) \left( \ln \psi_{u,v}(1, 0) - \ln \frac{y_u - y_{u,v}}{y_u(1 - y_v)} \right) \\ &\quad \quad + (y_v - y_{u,v}) \left( \ln \psi_{u,v}(0, 1) - \ln \frac{y_v - y_{u,v}}{(1 - y_u)y_v} \right) \\ &\quad \quad \left. + y_{u,v} \left( \ln \psi_{u,v}(1, 1) - \ln \frac{y_{u,v}}{y_u y_v} \right) \right], \end{aligned} \quad (5)$$

where  $-F$  is called the Bethe free energy function [26]. The gradient  $\nabla F(\mathbf{y}) = \left[ \frac{\partial F}{\partial y_v}, \frac{\partial F}{\partial y_{u,v}} \right]$  can be obtained as

$$\frac{\partial F}{\partial y_v} = \psi^{(v)} + \ln \frac{1 - y_v}{y_v} + \sum_{u \in \mathcal{N}(v)} \ln \left( \frac{1 - y_v - y_u + y_{u,v}}{1 - y_v} \cdot \frac{y_v}{y_v - y_{u,v}} \right) \quad (6)$$

$$\frac{\partial F}{\partial y_{u,v}} = \psi^{(u,v)} + \ln \left( \frac{y_u - y_{u,v}}{1 - y_u - y_v + y_{u,v}} \cdot \frac{y_v - y_{u,v}}{y_{u,v}} \right), \quad (7)$$

where

$$\psi^{(v)} := \ln \frac{\psi_v(1)}{\psi_v(0)} + \sum_{u \in \mathcal{N}(v)} \ln \frac{\psi_{u,v}(0,1)}{\psi_{u,v}(0,0)} \quad \text{and} \quad \psi^{(u,v)} := \ln \frac{\psi_{u,v}(0,0) \psi_{u,v}(1,1)}{\psi_{u,v}(1,0) \psi_{u,v}(0,1)}.$$

It is known that there is one-to-one correspondence between BP fixed points and zero gradient points of  $F$ . In particular, one can obtain the following lemma, where its proof can be done easily using the algebraic expressions (6) and (7) of gradients. The proof is presented in Appendix A.

**Lemma 1** *Given  $\varepsilon \in [0, 1]$ , suppose  $\mathbf{y} = [y_v, y_{u,v}]$  satisfies  $\|\nabla F(\mathbf{y})\|_\infty \leq \varepsilon$ . Then, the set of messages  $\{m_{u \rightarrow v}, m_{v \rightarrow u} : (u, v) \in E\}$  is a  $6\varepsilon$ -approximate BP fixed point if it is given as*

$$m_{u \rightarrow v} = \frac{\psi_{u,v}(0,1)}{\psi_{u,v}(0,0)} \cdot \frac{1 - y_v - y_u + y_{u,v}}{1 - y_v} \cdot \frac{y_v}{y_v - y_{u,v}}.$$

### 3 Algorithm for Computing BP Fixed Points

In this section, we present the main result of this paper, a new message passing algorithm for approximating a BP fixed point. From the (algebraic) relationship between approximate BP fixed points and near stationary points of the Bethe free energy function  $F$  in Lemma 1, it is equivalent to compute a near stationary point  $\mathbf{y}$  i.e.  $\|\nabla F(\mathbf{y})\|_2 \leq \varepsilon$ .

Our algorithm, described next, for finding such a point is essentially motivated by the standard (projected) gradient-descent algorithm. The non-triviality (and novelty) lies in our choice of appropriate (time-varying) ‘projection  $[\cdot]_*$ ’ with respect to the (time-varying) ‘step-size  $\frac{1}{\sqrt{t}}$ ’ at each iteration and subsequent analysis of rate of convergence.

#### Algorithm A

---

1. Algorithm parameters:

$$\varepsilon \in (0, 1) \quad \text{and} \quad \mathbf{y}(t) = [y_v(t) \in (0, 1) : v \in V] \quad \text{at the } t\text{-th iteration.}$$

2.  $\mathbf{y}(t)$  is updated as:

$$y_v(t+1) = \left[ y_v(t) + \frac{1}{\sqrt{t}} \left( \psi^{(v)} + \ln \frac{1 - y_v(t)}{y_v(t)} + \sum_{u \in \mathcal{N}(v)} \ln \left( \frac{1 - y_v(t) - y_u(t) + y_{u,v}(t)}{1 - y_v(t)} \cdot \frac{y_v(t)}{y_v(t) - y_{u,v}(t)} \right) \right) \right]_*,$$

where the projection  $[\cdot]_*$  at the  $t$ -th iteration is defined as

$$[x]_* = \begin{cases} x & \text{if } \frac{1}{t^{1/4}} \leq x \leq 1 - \frac{1}{t^{1/4}} \\ \frac{1}{t^{1/4}} & \text{if } x < \frac{1}{t^{1/4}} \\ 1 - \frac{1}{t^{1/4}} & \text{if } x > 1 - \frac{1}{t^{1/4}} \end{cases},$$

and  $y_{u,v}(t) > 0$  is computed as the unique solution satisfying

$$e^{\psi^{(u,v)}} \cdot \frac{y_u(t) - y_{u,v}(t)}{1 - y_u(t) - y_v(t) + y_{u,v}(t)} \cdot \frac{y_v(t) - y_{u,v}(t)}{y_{u,v}(t)} = 1 \quad \text{and} \quad y_{u,v}(t) < \min\{y_v(t), y_u(t)\}.$$

3. Compute messages  $\{m_{u \rightarrow v}, m_{v \rightarrow u}\}$  as

$$m_{u \rightarrow v} = \frac{\psi_{u,v}(0, 1)}{\psi_{u,v}(0, 0)} \cdot \frac{1 - y_v(t) - y_u(t) + y_{u,v}(t)}{1 - y_v(t)} \cdot \frac{y_v(t)}{y_v(t) - y_{u,v}(t)}.$$

4. Terminate if  $\{m_{u \rightarrow v}, m_{v \rightarrow u}\}$  is an  $\varepsilon$ -approximate BP fixed point.

---

The algorithm is clearly implementable through message-passing where each node  $u$  sends  $y_u(t)$  to all of its neighbors  $v \in \mathcal{N}(u)$  at each iteration. We also note that solving the second step for computing  $y_{u,v}(t)$  can be done efficiently since it is solving a quadratic equation whose coefficients are decided by  $y_v(t)$  and  $y_u(t)$ . We establish the following running time of the algorithm.

**Theorem 2** *Algorithm A terminates in  $2^{O(\Delta)} n^2 \varepsilon^{-4} \log^3(n \varepsilon^{-1})$  iterations as long as  $\psi_* = O(1)$ .*

The proof of Theorem 2 is presented in the following section. Note that the algorithm may require to maintain irrational messages or rational messages of long bits. In Section 3.2, we present a minor modification of the algorithm to fix the issue, which leads to a fully poly-time approximation algorithm (FPTAS) to compute an approximate BP fixed point.

### 3.1 Proof of Theorem 2

We first define  $F^*$  on  $(0, 1)^n$ : for  $\mathbf{y} = [y_v] \in (0, 1)^n$

$$F^*(\mathbf{y}) = F(\mathbf{y}, \mathbf{y}_E),$$

where  $F$  is the (original) Bethe free energy function defined in (5) and the additional vector  $\mathbf{y}_E = [y_{u,v}] \in (0, 1)^{|E|}$  is defined as the solution satisfying

$$\psi^{(u,v)} + \ln \left( \frac{y_u - y_{u,v}}{1 - y_u - y_v + y_{u,v}} \cdot \frac{y_v - y_{u,v}}{y_{u,v}} \right) = 0 \quad \text{and} \quad y_{u,v} < \min\{y_v, y_u\}. \quad (8)$$

Observe that each  $y_{u,v}$  is a function of  $y_u, y_v$ , i.e.  $y_{u,v} = y_{u,v}(y_u, y_v)$ . One can check that the gradient of  $F^*$  has the same form with that of  $F$  as follows.

$$\frac{\partial F^*}{\partial y_v} = \psi^{(v)} + \ln \frac{1 - y_v}{y_v} + \sum_{u \in \mathcal{N}(v)} \ln \left( \frac{1 - y_v - y_u + y_{u,v}}{1 - y_v} \cdot \frac{y_v}{y_v - y_{u,v}} \right), \quad \forall v \in V,$$

where we recall that  $y_{u,v}$  is decided in terms of  $y_u, y_v$  from (8). This implies that the updating procedure of  $\mathbf{y}(t)$  in the algorithm is simply as

$$\mathbf{y}(t+1) = \left[ \mathbf{y}(t) + \frac{1}{\sqrt{t}} \nabla F^*(\mathbf{y}(t)) \right]_* \quad (9)$$

Based on this interpretation, we start to prove the running time of the algorithm by stating the following key lemma.

**Lemma 3** *Define  $\delta > 0$  as the largest real number that satisfies the followings.*

$$\delta \leq \frac{1/2}{2(\Delta + 1)\psi_*^{4\Delta+1} + 1} \quad \text{and} \quad 4\delta \ln \frac{1}{2\delta} \leq 1.$$

Then,

$$\mathbf{y}(t) \in D := [\delta, 1 - \delta]^n, \quad \forall t \geq t_* := \delta^{-4}.$$



**Proof.** First observe that  $\mathbf{y}(t_*) \in D$  due to our choice of projection  $[\cdot]_*$  and  $\frac{1}{t_*^{1/4}} = \delta$ . Hence, it suffices to establish the following three steps: for all  $v \in V$  and  $t > t_*$ ,

$$\frac{\partial F^*}{\partial y_v} \leq 0 \quad \text{if } y_v \geq 1 - 2\delta \text{ and } \mathbf{y} \in D, \quad (10)$$

$$\frac{\partial F^*}{\partial y_v} \geq 0 \quad \text{if } y_v \leq 2\delta \text{ and } \mathbf{y} \in D, \quad (11)$$

$$\frac{1}{\sqrt{t}} \left| \frac{\partial F^*}{\partial y_v} \right| \leq \frac{\delta}{2} \quad \text{if } \mathbf{y} \in D. \quad (12)$$

From (9), (10), (11) and (12), it clearly follows that  $\mathbf{y}(t) \in D$  for all  $t \geq t_*$ .

**Proof of (10).** We first provide a proof of (10). To this end, if  $\mathbf{y} \in (0, 1)^n$ , we have

$$\begin{aligned} \frac{1 - y_v - y_u + y_{u,v}}{1 - y_v} \cdot \frac{y_v}{y_v - y_{u,v}} &= \frac{1}{1 + \frac{y_u - y_{u,v}}{1 - y_v - y_u + y_{u,v}}} \cdot \frac{y_v}{y_v - y_{u,v}} \\ &= \frac{1}{1 + e^{-\psi(u,v)} \cdot \frac{y_{u,v}}{y_v - y_{u,v}}} \cdot \left( 1 + \frac{y_{u,v}}{y_v - y_{u,v}} \right) \\ &\leq \max \left\{ 1, e^{\psi(u,v)} \right\} \\ &\leq \psi_*^4, \end{aligned} \quad (13)$$

where we use the definition (8) of  $y_{u,v}$ . Using this, (10) follows as

$$\begin{aligned} \frac{\partial F^*}{\partial y_v} &= \psi^{(v)} + \ln \frac{1 - y_v}{y_v} + \sum_{u \in \mathcal{N}(v)} \ln \left( \frac{1 - y_v - y_u + y_{u,v}}{1 - y_v} \cdot \frac{y_v}{y_v - y_{u,v}} \right) \\ &\leq \ln(2(\Delta + 1)\psi_*) + \ln \frac{2\delta}{1 - 2\delta} + \Delta \ln \psi_*^4 \\ &= \ln \frac{2(\Delta + 1)\psi_*^{4\Delta+1}}{\frac{1}{2\delta} - 1} \\ &\leq 0, \end{aligned}$$

where the last inequality is from our choice of  $\delta \leq \frac{1/2}{2(\Delta+1)\psi_*^{4\Delta+1}+1}$ .

**Proof of (11).** Second, we provide a proof of (11). Similarly as we did in (13), we have

$$\frac{1 - y_v - y_u + y_{u,v}}{1 - y_v} \cdot \frac{y_v}{y_v - y_{u,v}} \geq \min \left\{ 1, e^{\psi(u,v)} \right\} \geq \frac{1}{\psi_*^4}. \quad (14)$$

Hence, (11) follows as

$$\begin{aligned} \frac{\partial F^*}{\partial y_v} &= \psi^{(v)} + \ln \frac{1 - y_v}{y_v} + \sum_{u \in \mathcal{N}(v)} \ln \left( \frac{1 - y_v - y_u + y_{u,v}}{1 - y_v} \cdot \frac{y_v}{y_v - y_{u,v}} \right) \\ &\geq -\ln(2(\Delta + 1)\psi_*) + \ln \frac{1 - 2\delta}{2\delta} + \Delta \ln \frac{1}{\psi_*^4} \\ &= \ln \frac{\frac{1}{2\delta} - 1}{2(\Delta + 1)\psi_*^{4\Delta+1}} \\ &\geq 0, \end{aligned}$$

where the last inequality is again from our choice of  $\delta \leq \frac{1/2}{2(\Delta+1)\psi_*^{4\Delta+1}+1}$ .

**Proof of (12).** Finally, we provide a proof of (12). Using (13) and (14), it follows as

$$\begin{aligned}
\left| \frac{\partial F^*}{\partial y_v} \right| &= \left| \psi^{(v)} \right| + \left| \ln \frac{1-y_v}{y_v} \right| + \sum_{u \in \mathcal{N}(v)} \left| \ln \left( \frac{1-y_v-y_u+y_{u,v}}{1-y_v} \cdot \frac{y_v}{y_v-y_{u,v}} \right) \right| \\
&\leq \ln(2(\Delta+1)\psi_*) + \ln \frac{1-2\delta}{2\delta} + \Delta \ln \psi_*^4 \\
&= 2 \ln \frac{1-2\delta}{2\delta} \\
&\leq 2 \ln \frac{1}{2\delta} \\
&\leq \frac{\delta}{2} \cdot \sqrt{t_*},
\end{aligned}$$

where the last equality is from our choices of  $\delta, t_*$  which imply

$$\sqrt{t_*} = \delta^{-2} \geq 4 \frac{1}{\delta} \ln \frac{1}{2\delta}.$$

This completes the proof of Lemma 3.  $\square$

Using the above lemma, we will show that the algorithm terminates in  $2^{O(\Delta)} n^2 \varepsilon^{-4} \log^3(n \varepsilon^{-1})$  iterations until it outputs an  $\varepsilon$ -approximate BP fixed point. We first explain why it suffices to show the following:

$$\sum_{t=t_*}^T c_t \cdot \|\nabla F^*(\mathbf{y}(t))\|_2^2 = \frac{2^{O(\Delta)} n \log T}{\sqrt{T}}, \quad (15)$$

where  $c_t = \frac{t^{-1/2}}{\sum_{t=t_*}^T t^{-1/2}}$ . The above equality suggests that we can choose  $T = 2^{O(\Delta)} n^2 \varepsilon^{-4} \log^3(n \varepsilon^{-1})$  such that

$$\sum_{t=t_*}^T c_t \cdot \|\nabla F^*(\mathbf{y}(t))\|_2^2 \leq \left( \frac{\varepsilon}{6} \right)^2.$$

From  $\sum_{t=t_*}^T c_t = 1$ , there exists  $t \in [t_*, T]$  such that  $\|\nabla F^*(\mathbf{y}(t))\|_2 \leq \varepsilon/6$ . Further, observe that if  $\mathbf{y}_E(t)$  is defined from (8),

$$\|\nabla F(\mathbf{y}(t), \mathbf{y}_E(t))\|_2 \leq \|\nabla F^*(\mathbf{y}(t))\|_2 \leq \varepsilon/6.$$

Then, Lemma 1 implies that the computed messages at the  $t$ -th iteration is an  $\varepsilon$ -approximate BP fixed point.

Now we proceed toward establishing the desired inequality (15). The important implication of Lemma 3 is that the algorithm does not need the projection  $[\cdot]_*$  after the  $t_*$ -th iteration. In other words, from Lemma 3 and (9), we have that

$$\mathbf{y}(t+1) = \mathbf{y}(t) + \frac{1}{\sqrt{t}} \nabla F^*(\mathbf{y}(t)), \quad \forall t \geq t_*.$$

In what follows, we will assume  $t \geq t_*$  and  $\mathbf{y}(t) \in D$  from Lemma 3.

Using the Taylor's expansion, we have

$$\begin{aligned}
F^*(\mathbf{y}(t+1)) &= F^*\left(\mathbf{y}(t) + \frac{1}{\sqrt{t}} \nabla F^*(\mathbf{y}(t))\right) \\
&= F^*(\mathbf{y}(t)) + \nabla F^*(\mathbf{y}(t))' \cdot \frac{1}{\sqrt{t}} \nabla F^*(\mathbf{y}(t)) + \frac{1}{2} \frac{1}{\sqrt{t}} \nabla F^*(\mathbf{y}(t))' \cdot R \cdot \frac{1}{\sqrt{t}} \nabla F^*(\mathbf{y}(t)),
\end{aligned} \tag{16}$$

where  $R$  is a  $n \times n$  matrix such that

$$|R_{vw}| \leq \sup_{\mathbf{y} \in B} \left| \frac{\partial^2 F^*}{\partial y_v \partial y_w} \right|,$$

and  $B$  is a  $L_\infty$ -ball in  $\mathbb{R}^n$  centered at  $\mathbf{y}(t) \in D$  with its radius

$$r = \max_{v \in V} \left| \frac{1}{\sqrt{t}} \frac{\partial F^*}{\partial y_v}(\mathbf{y}(t)) \right|.$$

From (12), we know  $r \leq \frac{\delta}{2}$ . Hence,  $y_v \in [\delta/2, 1 - \delta/2]$  for  $\mathbf{y} = [y_v] \in B$ . Using this with  $\delta = 1/2^{O(\Delta)}$ , one can check that

$$\sup_{\mathbf{y} \in B} \left| \frac{\partial^2 F^*}{\partial y_v \partial y_w} \right| = \begin{cases} 2^{O(\Delta)} & \text{if } v = w, (v, w) \in E \\ 0 & \text{otherwise} \end{cases}.$$

Therefore, using these bounds the equality (16) reduces to

$$\begin{aligned}
F^*(\mathbf{y}(t+1)) &\geq F^*(\mathbf{y}(t)) + \frac{1}{\sqrt{t}} \|\nabla F^*(\mathbf{y}(t))\|_2^2 - \frac{O(2^{O(\Delta)}|E|)}{t} \\
&\geq F^*(\mathbf{y}(t)) + \frac{1}{\sqrt{t}} \|\nabla F^*(\mathbf{y}(t))\|_2^2 - \frac{O(2^{O(\Delta)}n)}{t},
\end{aligned}$$

since  $|E| \leq \Delta \cdot n$ . If we sum the above inequality over  $t$  from  $t_*$  to  $T-1$ , we have

$$F^*(\mathbf{y}(T)) \geq F^*(\mathbf{y}(t_*)) + \sum_{t=t_*}^{T-1} \frac{1}{\sqrt{t}} \|\nabla F^*(\mathbf{y}(t))\|_2^2 - O(2^{O(\Delta)}n) \sum_{t=t_*}^{T-1} \frac{1}{t}.$$

Since  $|F^*(\mathbf{y})| = O(\Delta n)$  for  $\mathbf{y} \in D$ , we obtain

$$\sum_{t=t_*}^{T-1} \frac{1}{\sqrt{t}} \|\nabla F^*(\mathbf{y}(t))\|_2^2 \leq O(\Delta n) + O(2^{O(\Delta)}n) \sum_{t=t_*}^{T-1} \frac{1}{t}.$$

Thus, we finally obtain the desired conclusion (15) as follows.

$$\begin{aligned}
\sum_{t=t_*}^T c_t \cdot \|\nabla F^*(\mathbf{y}(t))\|_2^2 &= \frac{1}{\sum_{t=t_*}^T \frac{1}{\sqrt{t}}} \sum_{t=t_*}^T \frac{1}{\sqrt{t}} \|\nabla F^*(\mathbf{y}(t))\|_2^2 \\
&\leq \frac{1}{\sum_{t=t_*}^T \frac{1}{\sqrt{t}}} \left( O(\Delta n) + O(2^{O(\Delta)}n) \sum_{t=t_*}^T \frac{1}{t} \right) \\
&= \frac{2^{O(\Delta)}n \log T}{\sqrt{T}},
\end{aligned}$$

where we recall that  $t_* = \delta^{-4} = 2^{O(\Delta)}$ . This completes the proof of Theorem 2.

### 3.2 Modification to FPTAS

In this section, we provide a minor modification of Algorithm A in Section 3, to establish a fully polynomial-time approximation scheme (FPTAS) for the BP fixed point computation. We will show only a polynomial number of bits are enough to maintain for each message  $y_v(t)$ . To this end, we define the following function  $g^t = [g_v^t]$  which describe the updating rule (at the  $t$ -th iteration) of Algorithm A, i.e.

$$\mathbf{y}(t+1) = g^t(\mathbf{y}(t)) \quad \text{under Algorithm A.}$$

Now we formally propose the following algorithm of minor modification.

#### Algorithm B

---

1. Algorithm parameters:

$$\varepsilon \in (0, 1) \quad \text{and} \quad \mathbf{z}(t) = [z_v(t) \in (0, 1) : v \in V] \quad \text{at the } t\text{-th iteration,}$$

where  $z_v(t)$  has  $k$ -bits (i.e.  $2^k z_v(t) \in \mathbb{Z}$ ) for all  $t \geq 0$ ,  $v \in V$ .

2.  $\mathbf{z}(t)$  is updated as:

$$\left| z_v(t+1) - g_v^t(\mathbf{z}(t)) \right| \leq \frac{1}{2^k}.$$

3. Compute a set of messages  $\{m_{u \rightarrow v}, m_{v \rightarrow u}\}$  satisfying

$$m_{u \rightarrow v} = \frac{\psi_{u,v}(0, 1)}{\psi_{u,v}(0, 0)} \cdot \frac{1 - z_v(t) - z_u(t) + z_{u,v}(t)}{1 - z_v(t)} \cdot \frac{z_v(t)}{z_v(t) - z_{u,v}(t)},$$

where  $z_{u,v}(t) > 0$  is computed to satisfy

$$\left| \psi^{(u,v)} + \ln \left( \frac{z_u(t) - z_{u,v}(t)}{1 - z_u(t) - z_v(t) + z_{u,v}(t)} \cdot \frac{z_v(t) - z_{u,v}(t)}{z_{u,v}(t)} \right) \right| \leq \frac{\varepsilon}{6}.$$

4. Terminate if  $\{m_{u \rightarrow v}, m_{v \rightarrow u}\}$  is an  $\varepsilon$ -approximate BP fixed point.
- 

We note that each step in the above algorithm is executable in a polynomial number of bitwise operations with respect to  $\Delta$ ,  $1/\varepsilon$ ,  $k$  and  $n$ . The second step to compute  $g_v^t$  consists of  $O(\Delta)$  arithmetic operations, logarithm, division, addition, square root and multiplication. Furthermore, the equations in the third step to compute  $\{m_{u \rightarrow v}, m_{v \rightarrow u}\}$  can be solvable in a polynomial number of bitwise operations with respect to  $1/\varepsilon$  and  $k$ .

Now we state the following theorem, which shows that one can choose  $k$  as a polynomial in terms of  $n$ ,  $1/\varepsilon$  and  $2^\Delta$ . This implies that Algorithm B is a FPTAS for such a choice of  $k$  as long as  $\Delta = O(\log n)$ . We note that one can obtain the explicit bound of  $k$  in terms of  $\Delta$ ,  $\psi_*$ ,  $n$  and  $\varepsilon$  via explicitly calculating each step in our proof.<sup>3</sup>

---

<sup>3</sup>Another naive way to avoid such an explicit choice of  $k$  is to run Algorithm B ‘polynomially’ many times by increasing  $k$  (as well as the number of iterations) until it succeeds.

**Theorem 4** For  $\psi_* = O(1)$ , there exists a  $k = 2^{O(\Delta)} n^2 \varepsilon^{-4} \log^4(n \varepsilon^{-1})$  such that Algorithm B terminates in  $2^{O(\Delta)} n^2 \varepsilon^{-4} \log^3(n \varepsilon^{-1})$  iterations.

**Proof.** To begin with, one can check that  $g^t$  is  $2^{O(\Delta)}$ -Lipschitz in  $[\delta/2, 1 - \delta/2]^n$  where  $\delta = 1/2^{O(\Delta)}$  is the constant in Lemma 3. Formally speaking, for all  $t \geq 0$ ,  $\mathbf{y}_1, \mathbf{y}_2 \in [\delta/2, 1 - \delta/2]^n$ ,

$$\|g^t(\mathbf{y}_1) - g^t(\mathbf{y}_2)\|_1 \leq L \cdot \|\mathbf{y}_1 - \mathbf{y}_2\|_1,$$

where  $L = 2^{O(\Delta)}$ . Let  $\mathbf{y}(t)$  and  $\mathbf{z}(t)$  are parameters of Algorithm A and B, respectively. Initially, suppose  $\mathbf{y}(0) = \mathbf{z}(0)$ . Then, if  $\mathbf{y}(t), \mathbf{z}(t) \in [\delta/2, 1 - \delta/2]^n$ ,

$$\begin{aligned} \|\mathbf{y}(t+1) - \mathbf{z}(t+1)\|_1 &\leq \|g^t(\mathbf{y}(t)) - g^t(\mathbf{z}(t))\|_1 + \frac{n}{2^k} \\ &\leq L \cdot \|\mathbf{y}(t) - \mathbf{z}(t)\|_1 + \frac{n}{2^k} \\ &= h(\|\mathbf{y}(t) - \mathbf{z}(t)\|_1), \end{aligned} \tag{17}$$

where we define  $h(x) := L \cdot x + \frac{n}{2^k}$ . From Lemma 3, we know  $\mathbf{y}(t) \in [\delta, 1 - \delta]^n$  for all  $t \geq 0$ . Thus, (17) holds for  $t \geq 0$  with  $h^{(t)}(0) \leq \delta/2$ .<sup>4</sup> Further, using  $h^{(t)}(0) < (L+1)^t \cdot n/2^k$ , it follows that

$$\|\mathbf{y}(t) - \mathbf{z}(t)\|_1 \leq \gamma, \quad \text{for } t \leq T := \frac{\log(2^k \gamma / n)}{\log(L+1)}, \tag{18}$$

where  $\gamma < \delta/2$  will be decided later.

Now it is not hard to see that for  $t \leq T$  (i.e.  $\mathbf{y}(t), \mathbf{z}(t) \in [\delta/2, 1 - \delta/2]^n$  and  $\|\mathbf{y}(t) - \mathbf{z}(t)\|_1 \leq \gamma$ ), if  $\|\nabla F^*(\mathbf{y}(t))\|_\infty \leq \gamma$ , then  $\|\nabla F^*(\mathbf{z}(t))\|_\infty \leq \gamma \cdot 2^{O(\Delta)}$ . Observe that one can choose  $\gamma = \varepsilon/2^{O(\Delta)}$  and  $k = 2^{O(\Delta)} n^2 \gamma^{-4} \log^4(n \gamma^{-1})$  so that Algorithm A has  $\|\nabla F^*(\mathbf{y}(t))\|_\infty \leq \gamma$  in  $T$  iterations, and hence Algorithm B has  $\|\nabla F^*(\mathbf{z}(t))\|_\infty \leq \varepsilon/6$  in the same number of iterations. Therefore, the conclusion of Theorem 4 follows from Lemma 1.  $\square$

## 4 Extension to Non-binary Graphical Models

In this section, we discuss how to design a similar algorithm to those in the previous section for non-binary graphical models. Here we provide a high-level description, but one can check the further details based on the identical arguments to the binary case in Section 3.

Consider non-binary random variables  $\{x_v\}$  in the graphical model described in Section 2, i.e.,  $x_v \in [Q] = \{0, 1, \dots, Q\}$  for some  $Q \in \mathbb{N}$ . Hence, the potential functions  $\psi_{u,v}$  and  $\psi_v$  are functions on  $[Q]^2$  and  $[Q]$ , respectively. We remind the reader that the essential goal is to find a near stationary point of the following Bethe approximation under the constraints  $\sum_{x_v} \tau_v(x_v) = 1$  and  $\tau_v(x_v) = \sum_{x_u} \tau_{u,v}(x_u, x_v)$ .

$$\sum_{v \in V} \sum_{x_v} \tau_v(x_v) [\ln \psi_v(x_v) - \ln \tau_v(x_v)] + \sum_{\{u,v\} \in E} \sum_{x_u, x_v} \tau_{u,v}(x_u, x_v) \left[ \ln \psi_{u,v}(x_u, x_v) - \ln \frac{\tau_{u,v}(x_u, x_v)}{\tau_u(x_u) \tau_v(x_v)} \right].$$

First, for every pair  $(p, q) \in [Q]^2$ , one can define  $F_{p,q}^*$  on  $\mathbf{y} = [y_v] \in [0, 1]^n$  similar to  $F^*$  in Section 3.1, by (a) fixing variables except for  $\{\tau_v(p), \tau_v(q) : v \in V\}$  and  $\{\tau_{u,v}(p, q) : (u, v) \in E\}$ ,

<sup>4</sup> $h^{(t)}$  is the function composing  $h$  't times', i.e.  $h^{(t)} = h \circ h^{(t-1)}$  and  $h^{(1)} = h$ .

(b) setting  $y_v = \tau_v(p)$ , and (c) considering the constraints  $\sum_{x_v} \tau_v(x_v) = 1$  and  $\tau_v(x_v) = \sum_{x_u} \tau_{u,v}(x_u, x_v)$ . Now the algorithm for a non-binary graphical model maintains variables

$$\{\tau_v^t(x_v) : v \in V, x_v \in [Q]\} \quad \text{and} \quad \{\tau_{u,v}^t(x_v, x_u) : (u, v) \in E, (x_v, x_u) \in [Q]^2\},$$

at the  $t$ -th iteration. At each round, it pick a pair  $(p, q) \in [Q]^2$  and updates the vector  $\{\tau_v^t(p)\}$  as

$$\tau_v^{t+1}(p) \leftarrow \tau_v^t(p) + \frac{\alpha}{\sqrt{t}} \nabla F_{p,q}^* (\{\tau_u^t(p) : u \in V\}),$$

for some constant  $\alpha > 0$ .  $\{\tau_v^t(q) : v \in V\}$  and  $\{\tau_{u,v}^t(p, q) : (u, v) \in E\}$  can be also updated according to  $\{\tau_v^t(p) : v \in V\}$  so that they satisfy the constraints  $\sum_{x_v} \tau_v(x_v) = 1$  and  $\tau_v(x_v) = \sum_{x_u} \tau_{u,v}(x_u, x_v)$  as well as quadratic equations corresponding to (8) of the binary case. From similar arguments in the proof of Lemma 3, one can show that it is possible to choose  $\alpha$  small enough so that updated marginal probabilities  $\{\tau_v^{t+1}(p), \tau_v^{t+1}(q) : v \in V\}$  and  $\{\tau_{u,v}^{t+1}(p, q) : (u, v) \in E\}$  are always valid (i.e., positive). And hence, the convergence rate analysis follows as we did in Section 3.1. We omit the further details in this paper.

## 5 Conclusion

In the last decade, exciting progresses have been made on understanding computationally hard problems in computer science using a variety of methods in statistical physics. The belief propagation (BP) algorithm or its variants are on this line and suggest to solve certain ‘relaxations’ of hard problems. In this paper, we address the question whether the relaxation is indeed computationally easy to solve in a strong sense. We believe that our rigorous complexity analysis of the BP-relaxation is the important step to guarantee the complexity of BP-based algorithms.

## 6 Acknowledgement

We are grateful to Devavrat Shah for fruitful comments, revisions and encouragement on this paper. We are also grateful to Max Welling for pointing out the similarity between our algorithm and that in [23].

## References

- [1] A. Bandyopadhyay and D. Gamarnik. Counting without sampling: new algorithms for enumeration problems using statistical physics. *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, 890-899, 2006.
- [2] H. A. Bethe. Statistical theory of superlattices. *Proc. Roy. Soc. London A*, 150:552-558, 1935.
- [3] C. Berrou, A. Glavieux and P. Thitimajshima. Near Shannon Limit Error-correcting Coding and Decoding: Turbo codes (I). *Proceeding of ICC (Geneva)*, 1993.
- [4] V. Chandrasekaran, M. Chertkov, D. Gamarnik, D. Shah and J. Shin. Counting independent sets using the Bethe approximation. *SIAM Journal on Discrete Mathematics*, 25(2):1012-1034, 2011.

- [5] V. Chandrasekaran, N. Srebro and P. Harsha. Complexity of inference in graphical models, *Uncertainty in Artificial Intelligence*, 2008.
- [6] M. Chertkov and V. Y. Chernyak. Loop series for discrete statistical models on graphs. *Journal of Statistical Mechanics: Theory and Experiment*, 2006.
- [7] C. Daskalakis and C. Papadimitriou. Continuous Local Search. *Proceedings of ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2011.
- [8] A. Dembo and A. Montanari. Ising models on locally tree-like graphs. *The Annals of Applied Probability*, 20(2): 565-592, 2010.
- [9] C. Domb and M. S. Green. Phase Transitions and Critical Phenomena. *Vol. 2. Academic Press. London*, 1972.
- [10] C. D. Forney, Jr. Codes on Graphs: News and Views. *Conference on Information Sciences and Systems*. The John Hopkins University, 2001.
- [11] D. A. Forsyth, J. Haddon and S. Ioffe. The Joy of Sampling. *International Journal of Computer Vision*, 41(1):109-134, 2001.
- [12] W. T. Freeman and E. C. Pasztor. Learning Low Level Vision. *In Proceeding of International Conference of Computer Vision*, 1999.
- [13] D. S. Gaunt and M. E. Fisher. Hard-Sphere Lattice Gases. I. Plane-Square Lattice. *Journal of Chemical Physics*, 43(8):2840-2863, 1965.
- [14] T. Heskes. On the uniqueness of loopy belief propagation fixed points. *Neural Computation*, 16(11):2379. 2413, 2004.
- [15] A. T. Ihler, J. W. Fischer III and A. S. Willsky. Loopy Belief Propagation: Convergence and Effects of Message Errors. *The Journal of Machine Learning Research*, 6:905-936, 2005
- [16] S. L. Lauritzen. *Graphical models*. Oxford University Press, USA, 1996.
- [17] R. J. McEliece, D. J. C. Mackay and J. F. Cheng. Turbo decoding as an instance of Pearl’s belief propagation algorithm. *IEEE Journal on Selected Areas in Communication*, 16(2):140-152, 1998.
- [18] K. P. Murphy, Y. Weiss and M. Jordan. Loopy belief propagation for approximate inference: an empirical study. *In Proceedings of Uncertainty in Artificial Intelligence*, 1999.
- [19] J. Pearl. Reverend Bayes on inference engines: A distributed hierarchical approach. *Proceedings of the Second National Conference on Artificial Intelligence*, AAAI Press, USA, 1982.
- [20] F. Ricci-Tersenghi and G. Semerjian. On the cavity method for decimated random constraint satisfaction problems and the analysis of belief propagation guided decimation algorithms. *J. Stat. Mech.*, 2009
- [21] E. B. Sudderth, M. J. Wainwright, and A. S. Willsky. Loop series and Bethe variational bounds in attractive graphical models. *Advances in neural information processing systems*, 20:1425–1432, 2008.

- [22] S. Tatikonda and M. Jordan. Loopy belief propagation and gibbs measures. *Uncertainty in Artificial Intelligence*, 2002.
- [23] Y. W. Teh and M. Welling. Belief Optimization for Binary Networks: A stable Alternative to Loopy Belief Propagation. *Uncertainty in Artificial Intelligence*, 2001.
- [24] M. J. Wainwright, T. Jaakkola and A. S. Willsky. Tree-based reparameterization framework for analysis of sum-product and related algorithms. *IEEE Transactions on Information Theory*, 45(9):1120-1146, 2003.
- [25] Y. Weiss. Correctness of local probability propagation in graphical models with loops. *Neural Computation*, 12(1), 2000.
- [26] J. Yedidia, W. Freeman, and Y. Weiss. Constructing free energy approximations and generalized belief propagation algorithms. *IEEE Transactions on Information Theory*, 51:2282-2312, 2004.
- [27] A. L. Yuille. CCCP algorithms to minimize the Bethe and Kikuchi free energies: Convergent alternatives to belief propagation. *Neural Computation*, 14, 2002.



## A Proof of Lemma 1

To simplify notations, we use  $a = b \cdot e^{\pm\varepsilon}$  to mean  $b \cdot e^{-\varepsilon} \leq a \leq b \cdot e^\varepsilon$ . Now from  $\left| \frac{\partial F}{\partial y_v} \right| \leq \varepsilon$  and  $\left| \frac{\partial F}{\partial y_{u,v}} \right| \leq \varepsilon$ , we have

$$\frac{\psi_u(1)}{\psi_u(0)} \cdot \frac{\psi_{u,v}(1,0)}{\psi_{u,v}(0,0)} \cdot \frac{1 - y_v - y_u + y_{u,v}}{y_u - y_{u,v}} \cdot \prod_{w \in \mathcal{N}(u) \setminus v} m_{w \rightarrow u} = e^{\pm\varepsilon}.$$

$$e^{\psi(u,v)} \cdot \frac{y_u - y_{u,v}}{1 - y_u - y_v + y_{u,v}} \cdot \frac{y_v - y_{u,v}}{y_{u,v}} = e^{\pm\varepsilon}.$$

Using the above inequalities, the desired conclusion of Lemma 1 follows as

$$\begin{aligned} m_{u \rightarrow v} &= \frac{\psi_{u,v}(0,1)}{\psi_{u,v}(0,0)} \cdot \frac{1 - y_v - y_u + y_{u,v}}{1 - y_v} \cdot \frac{y_v}{y_v - y_{u,v}} \\ &= \frac{\psi_{u,v}(0,1)}{\psi_{u,v}(0,0)} \cdot \frac{1 + \frac{y_{v,u}}{y_v - y_{u,v}}}{1 + \frac{y_u - y_{u,v}}{1 - y_v - y_u + y_{u,v}}} \\ &= \frac{\psi_{u,v}(0,1)}{\psi_{u,v}(0,0)} \cdot \frac{1 + e^{\pm\varepsilon} \cdot e^{\psi(u,v)} \cdot \frac{y_u - y_{u,v}}{1 - y_u - y_v + y_{u,v}}}{1 + \frac{y_u - y_{u,v}}{1 - y_v - y_u + y_{u,v}}} \\ &= \frac{\psi_{u,v}(0,1)}{\psi_{u,v}(0,0)} \cdot \frac{1 + e^{\pm 2\varepsilon} \cdot e^{\psi(u,v)} \cdot \frac{\psi_u(1)}{\psi_u(0)} \cdot \frac{\psi_{u,v}(1,0)}{\psi_{u,v}(0,0)} \cdot \prod_{w \in \mathcal{N}(u) \setminus v} m_{w \rightarrow u}}{1 + e^{\pm\varepsilon} \cdot \frac{\psi_u(1)}{\psi_u(0)} \cdot \frac{\psi_{u,v}(1,0)}{\psi_{u,v}(0,0)} \cdot \prod_{w \in \mathcal{N}(u) \setminus v} m_{w \rightarrow u}} \\ &= e^{\pm 3\varepsilon} \cdot \frac{\psi_{u,v}(0,1)}{\psi_{u,v}(0,0)} \cdot \frac{1 + e^{\psi(u,v)} \cdot \frac{\psi_u(1)}{\psi_u(0)} \cdot \frac{\psi_{u,v}(1,0)}{\psi_{u,v}(0,0)} \cdot \prod_{w \in \mathcal{N}(u) \setminus v} m_{w \rightarrow u}}{1 + \frac{\psi_u(1)}{\psi_u(0)} \cdot \frac{\psi_{u,v}(1,0)}{\psi_{u,v}(0,0)} \cdot \prod_{w \in \mathcal{N}(u) \setminus v} m_{w \rightarrow u}} \\ &= e^{\pm 3\varepsilon} \cdot f_{u \rightarrow v} \left( \prod_{w \in \mathcal{N}(u) \setminus v} m_{w \rightarrow u} \right) \\ &= (1 \pm 6\varepsilon) \cdot f_{u \rightarrow v} \left( \prod_{w \in \mathcal{N}(u) \setminus v} m_{w \rightarrow u} \right). \end{aligned}$$